普通高等教育“十三五”规划教材

Visual Basic程序设计

主　编　王伟宇
副主编　邵利平　程　灏　刘汉强　强　雪

科学出版社

普通高等教育“十三五”规划教材

Visual Basic 程序设计

主编　王伟宇

副主编　邵利平　程　灏　刘汉强　强　雪

科学出版社

北京

内 容 简 介

本书针对非计算机专业学生的特点，以知识点为主线，以大量实例为辅助的方式，系统地介绍了 Visual Basic 相关知识。全书内容包括引言、可视化编程基础、Visual Basic 语言基础、程序流程控制、过程、数组、结构和类、用户界面设计初步、文件操作、数据库编程、程序调试等。本书是 Visual Basic 程序设计的入门教材，采用 Visual Basic 2010 版本，也适合于更高的版本。本书内容丰富，语言叙述通俗易懂，注重理论与实践相结合。

本书既可作为高等院校非计算机专业程序设计基础课教材，也可作为广大计算机爱好者的自学教材。

图书在版编目(CIP)数据

Visual Basic 程序设计/王伟宇主编. —北京: 科学出版社, 2017. 1
普通高等教育“十三五”规划教材

ISBN 978-7-03-051340-3

Ⅰ. ①V… Ⅱ. ①王… Ⅲ. ①BASIC 语言-程序设计-高等学校-教材
Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字 (2017) 第 003945 号

责任编辑: 李 萍 周 莎 / 责任校对: 钟 洋
责任印制: 张 倩 / 封面设计: 铭轩堂

科学出版社 出版
北京东黄城根北街 16 号
邮政编码: 100717
http://www.sciencep.com

保定市中画美凯印刷有限公司 印刷
科学出版社发行 各地新华书店经销
*
2017 年 1 月第 一 版 开本: 720×1000 1/16
2017 年 1 月第一次印刷 印张: 25
字数: 510 000

定价: 65.00 元

(如有印装质量问题, 我社负责调换)

前　言

进入信息时代以来，计算机的应用就深入到人们日常工作生活中，高等院校计算机基础课程的设置与社会对人才所具备计算机应用技能的需求是否接轨显得尤为重要。如今，全国大多数高校面向非计算机专业学生开设了 C、Visual Basic、Java 等程序设计语言，其中 C 语言尤其占多数。但 C 语言具有不支持面向对象技术和非图形界面等缺点，对于很多缺乏一定预备知识的非计算机专业学生而言，学习上有一定难度。

Visual Basic 是微软公司于 1991 年发布的结构化的、模块化的、面向对象的、包含协助开发环境的事件驱动为机制的可视化程序设计语言。在发展的过程中，经过多次版本的升级，从最早的 1.0 版本到拥有广大用户的 6.0 版本，Visual Basic 逐渐成为非常成熟和稳定的开发系统。随着微软.NET 计划的实施，Visual Basic 语言迎来了一次大的变革，2002 年推出的新的版本命名为 Visual Basic.NET，新增了许多特性和语法，将 Visual Basic 语言推上一个新的高度。从 2005 年开始，Visual Basic 的版本号使用年份命名，并去掉了“.NET”字样。到目前为止，最近的版本是 Visual Basic 2015。

与 C 语言等程序设计语言相比，Visual Basic 完全支持面向对象技术，具有简单而高效的可视化程序设计界面，语法简单严谨，非常适合作为程序设计的入门语言。为此，我们结合多年一线教学的经验和学生的反馈，编写本书，力图将学生逐步引入到程序设计的世界中，进而培养他们分析问题和解决问题的能力。由于当下主流操作系统是 Windows7 及以上版本，考虑到兼容性问题，本书用 Visual Basic 2010 版本进行编写，采用完全免费的 Visual Basic 2010 学习版开发环境，同时也适合更高版本的入门学习。

作为程序设计入门教材，本书体现了以下特点：

(1) 不需要学习者有任何编程经验，可以作为第一门计算机语言来学习。

(2) 从最简单的窗口程序设计入手，逐步引入数据类型、流程控制、模块化思想、面向对象技术等主线知识，由浅入深，循序渐进，符合普遍的认知规律。

(3) 强化算法的概念，让学习者感受从分析问题到解决问题的完整过程。

(4) 例题多且设计合理，多数例题从题目分析着手，到窗口设计和代码实现，过程完整，有利于引导学生自主学习。

(5) 将常用的函数和控件集成到附录中，便于学习者查阅。

(6) 每章设计有思考题和实验，以巩固相应知识。

本书目录框架的制订和统稿工作由王伟宇完成，第 1、2 章和附录 2~4 由王伟宇编写，第 3、9 章由刘汉强编写，第 4、10 章由程灏编写，第 5、6 章由邵利平编写，第 7、8 章和附录 1 由强雪编写。本书中带“*”内容为选学内容。

感谢刘侍刚教授对本书编写的关心及支持，感谢所有为本书出版付出心血的人员。

感谢使用本书的读者，由于作者水平有限，书中难免有不足之处，欢迎各位读者批评指正。

目　录

第1章 引 言

1.1 Visual Basic 简介

1.1.1 计算机语言和程序

日常生活中，人与人之间的交流需要语言来进行，而人和计算机之间的交流，也需要语言来进行，即需要计算机语言来交流。与人类的语言相似，计算机语言就是一套具有数字、字符、词法和语法规则的系统，由这些规则组成计算机各种指令(或各种语句)，实现人与计算机之间的通讯。

那么，计算机语言又是如何实现计算机和人之间沟通的呢？这就是人和计算机之间的“中间人”——程序的功劳了。通过计算机语言(如常见的汇编语言、C 语言和 Basic 语言等)编制程序，在程序中就可以表达我们的意图，而程序则负责调度各种计算机资源(申请内存和执行计算等)来完成我们下达的命令。要通过程序向计算机传达一个命令，必须经过下面的步骤。

首先，在源文件中用程序设计语言表达指令。用户通过计算机语言表达指令，通常，指令是记录在文件中以源代码的形式出现的。在进行具体操作的时候，根据用户所使用的开发平台，具体的操作步骤可能有所不同。

然后，将源代码编译成可执行文件。在源文件中记录的指令通常是以高级程序设计语言表达的，这种语言接近于人类使用的自然语言。但是，对于计算机来说，这些符号是无法理解的，因此需要一个来完成这个“翻译”工作的角色——编译器。它负责将源代码文件中以高级程序设计语言表达的指令翻译成计算机可以理解的机器语言，并记录在可执行文件中(在 Windows 开发平台下，可执行程序通常是一个扩展名为.exe 的文件)。经过这个编译过程后，就得到了可执行的文件，也就使源代码文件变为了类似于 11000100111000011010111100000111000000011110001 这样的二进制序列。

最后，计算机执行可执行程序。这些二进制指令会被调入计算机，并遵照这些指令执行，完成相关任务。

1.1.2　从机器语言到汇编语言

计算机工作基于二进制，从根本上说，计算机只能识别由 1 和 0 组成的指令。在计算机发展的初期，一般计算机的指令长度为 16 位，即以 16 个二进制数(0 或 1)组成一条指令。16 个 0 和 1 可以组成各种排列组合。例如，用 1011011000000000 让计算机进行一次加法运算。用户想要计算机知道和执行自己的意图，就要编写许多条由 1 和 0 组成的指令，然后用纸袋穿孔机以人工的方法在特质的黑色纸带上穿孔，在指定的位置上有孔代表 1，无孔代表 0。一个程序往往需要一个长长的纸带，在需要运行此程序时就将纸带装在光电输入机上，当光电输入机从纸带读入信息时，有孔处产生一个电脉冲，指令变成电信号，让计算机执行各种操作。

这种计算机能直接识别和接受的二进制代码称为机器指令(machine instruction)，机器指令的集合就是该计算机的机器语言(machine language)，在语言的规则中规定各种指令的表现形式和作用。

显然，机器语言与人们习惯用的语言差别太大，难学、难写、难记、难检查、难修改，且难以推广使用，因此初期只有极少数的计算机专业人员会编写这样的计算机程序。

为了克服机器语言的上述缺点，人们创造出符号语言(symbolic language)，它用一些英文字母和数字表示一个指令，如 ADD 代表“加”，SUB 代表“减”，LD 代表“传送”等。例如，两个数相加的机器指令可以用下面符号指令代替：

ADD A,B　(执行 A+B=>A，将寄存器中 A 的数与寄存器 B 中的数相加，放到寄存器 A 中)

显然，计算机并不能直接识别和执行符号语言的指令，需要用一种称为汇编程序的软件，把符号语言的指令转换为机器指令。一般地，一条符号语言的指令对应转换为一条机器指令，转换的过程称为“代真”或“汇编”语言，因此符号语言又称为符号汇编语言(symbolic assembler language)或汇编语言(assembler language)。

1.1.3　从汇编语言到高级语言

虽然汇编语言比机器语言简单好记一些，但仍然难以普及，只在专业人员中使用。不同型号计算机的机器语言和汇编语言是互不通用的，即用甲机器的机器语言编写的程序在乙机器上不能使用。机器语言和汇编语言是完全依赖于具体机器特征的，是面向机器的语言。由于它“贴近”计算机，或者说离计算机“很近”，因此称为计算机低级语言(low level language)。

为了克服低级语言的缺点，20 世纪 50 年代创造出了第一个计算机高级语言——FORTRAN 语言。它很接近于人们习惯使用的自然语言和数学语言。程序中用到的语句和指令使用英文单词表示，所用的运算符和运算表达式与人们日常所用的数学式子差不多，很容易理解。程序运行的结果用英文和数字输出，十分方便。例如，在 FORTRAN 语言程序中，若要输出 $3.5\times 6\sin\left(\frac{\pi}{3}\right)$，只需写出下面这样一个语句：

```
PRINT*,3.5*6*sin(3.1415926/3)
```

即可得到计算结果，显然这是很容易理解和使用的。

这种语言功能很强，且不依赖于具体机器，用它写出的程序对任何型号的计算机都适用(或只需做很少的修改)。它与具体机器距离较远，故称为计算机高级语言(high level language)。

当然，计算机是不能直接识别高级语言程序的，首先要进行“翻译”，用一种称为编译程序的软件把用高级语言写的程序(称为源程序，source program)转换为机器指令的程序(称为目标程序，object program)，然后让计算机执行机器指令程序，最后得到结果。高级语言的一个语句往往对应多条机器指令。

自从有了高级语言后，一般的科技人员、管理人员、大中学生以及计算机爱好者，都能较容易地学会用高级语言编写程序，指挥计算机进行工作，而完全不顾什么机器指令，也可以不必深入懂得计算机的内部结果和工作原理，就能得心应手地利用计算机进行各种工作，为计算机的推广普及创造了良好的条件。

数十年来，全世界涌现了 2500 种以上高级语言，每种高级语言都有其特定的用途，其中应用较广泛的有 100 多种，影响较大的有 FORTRAN 和 ALGOL(适合数值计算)、BASIC 和 QBASIC(适合初学者的小型会话语言)、COBOL(适合商业管理)、Pascal(适合数学的结构程序设计语言)、PL/1(大型通用语言)、LISP 和 PROLOC(人工智能语言)、C(系统描述语言)、C++(支持面向对象程序设计的大型语言)、Visual Basic(支持面向对象程序设计的语言)和 Java(适于网络的语言)等。

1.1.4　Visual Basic 的发展史

BASIC 语言属于高级语言的一种，英文名称的全名是“Beginner’s All-Purpose Symbolic Instruction Code”，取其首字字母简称“BASIC”，就名称的含意来看，是适用于“初学者的多功能符号指令码”。最初，它是为初学计算机的人员设计的，作为交互计算的第一个语言，且附加了很多特征，简单易学而且能适应各种程序设计任务。

到了 20 世纪 70 年代，BASIC 语言得到了较全面的完善并能处理相当复杂的应用问题。BASIC 语言在早期发展过程中有很多版本，其中微软公司于 1987 年推出 MSDOS 环境下的 Quick BASIC，克服了原 BASIC 结构化程度差和运算速度慢的缺点，具有与 FORTRAN 和 C 语言相媲美的完善程度，有方便全面的编程环境和库管理功能，可以说 Quick BASIC 的易学性和 BASIC 差不多，而功能则大为扩充，因此得到了广泛好评。

从 20 世纪 80 年代末到 90 年代初，随着 Windows 操作系统的发展和逐步流行，PC 的操作方式由命令行的方式向图形用户界面方式转变。Cooper 公司总裁 Alan Cooper 开始开发一个新的开发环境，在这个开发环境中，甚至连基于窗口的用户界面的屏幕布局都用鼠标来驱动，其目标是针对那些为产生一个用户界面连代码都懒得写的程序员。因此，他决定这个产品使用的语言应该是 BASIC 语言的一个变种。1991 年，在 Alan Cooper 的工作成果基础上，微软公司推出了名为 Visual Basic 的产品，版本号为 1.0。程序员可以安装它，并能立即写出 Windows 界面程序。虽然该产品功能很少，但是它首次集成了可视化的程序界面设计和事件驱动机制，具有划时代的意义。

在 Windows 操作系统的不断发展和成熟中，Visual Basic 也在不断地发展中。到 1998 年时，Visual Basic 发展到了 6.0 的版本。在版本的变化中逐步增加了处理关系数据库的功能，也增加了对类和自建控件的支持，并开始支持中文等，最终成为一个非常稳定、强健的开发工具，至今仍然拥有数以百万计的用户。

随着微软提出的.NET 框架思想，在 1999 年，微软公司决定将 Visual Basic 改成真正的面向对象的语言，并于 2002 年在.NET 平台中正式推出 Visual Basic.NET，2005 年推出 Visual Basic 2005 版本。值得注意的是从这个版本开始去掉了“.NET”字样，而且这个版本与前面的版本相比有了很多重大的变革，增加了 My 伪命名空间、泛型和操作符重载等众多新特性。其后几年间微软公司发布了若干个版本，直到最近的 Visual Basic 2015 版本。

本书是基于 Visual Basic 2010 版本(书中统称 Visual Basic)编写的。

1.2　Visual Basic 开发环境

1.2.1　.NET 框架

.NET 框架由微软开发，是一个致力于敏捷软件开发、快速应用开发、平台无关性和网络透明化的软件开发平台，其中包含了许多有助于互联网和内部网应用迅

捷开发的技术。.NET 框架是以一种采用系统虚拟机运行的编程平台，支持多种语言(C#、Visual Basic、C++和 Python 等)的开发。.NET 也为应用程序接口(Application Programming Interface，API)提供了新功能和开发工具。这些革新使得程序设计员可以同时进行 Windows 应用软件和网络应用软件以及组件和服务的开发。.NET 框架的结构如图 1-1 所示。

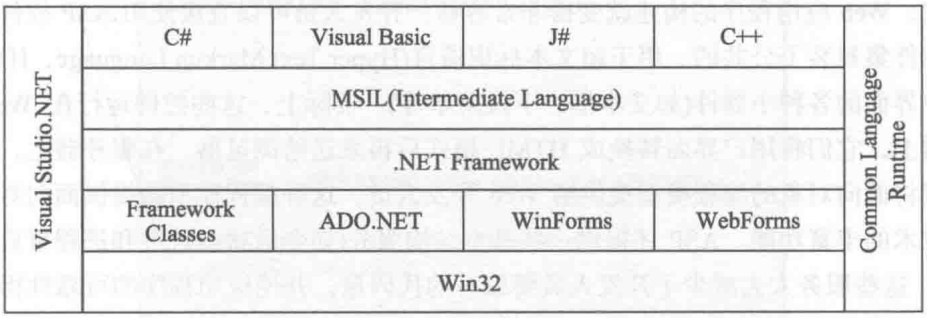

图 1-1　.NET 框架

.NET 框架是一个多语言组件开发和执行环境，它由以下三个主要部分组成。

1. 公共语言运行时

公共语言运行时(Common Language Runtime，CLR)名称不能准确反映它的全部功能。实际上，公共语言运行时在组件的开发及运行过程中，都扮演着非常重要的角色。在组件运行过程中，运行时负责管理内存分配、启动或删除线程和进程、实施安全性策略，同时满足当前组件对其他组件的需求。在开发阶段，运行时的作用有些变化：与现今的组件对象模型(Component Object Model，COM)相比，运行时的自动化程度大为提高(如可自动执行内存管理)，因此开发人员的工作变得非常轻松。.NET 框架的关键作用在于，它提供了一个跨编程语言的统一编程环境，这也是它能独树一帜的根本原因。

2. 统一的编程类

.NET 框架为开发人员提供了一个统一、面向对象、层次化和可扩展的类库集(API)。现今，C++开发人员使用的是 Microsoft 基类库，Java 开发人员使用的是 Windows 基类库，而 Visual Basic 用户使用的又是 Visual Basic API 集，只要简单地使用.NET 框架就可以统一微软当前的各种不同类框架。这样，开发人员无需学习多种框架就能顺利编程。远不止于此的是，通过创建跨编程语言的公共类库集，.NET 框架可实现跨语言继承性、错误处理功能和调试功能。实际上，从